

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

① **BLACK BORDERS**

- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-229060

(P2001-229060A)

(43) 公開日 平成13年8月24日 (2001.8.24)

(51) Int.Cl.⁷

G 0 6 F 12/00

識別記号

5 2 0

F I

G 0 6 F 12/00

テ-マコ-ト* (参考)

5 2 0 A 5 B 0 8 2

審査請求 有 請求項の数18 O L (全 25 頁)

(21) 出願番号

特願2000-39682(P2000-39682)

(22) 出願日

平成12年2月17日 (2000.2.17)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 安村 義孝

東京都港区芝五丁目7番1号 日本電気株式会社社内

(74) 代理人 100088328

弁理士 金田 暢之 (外2名)

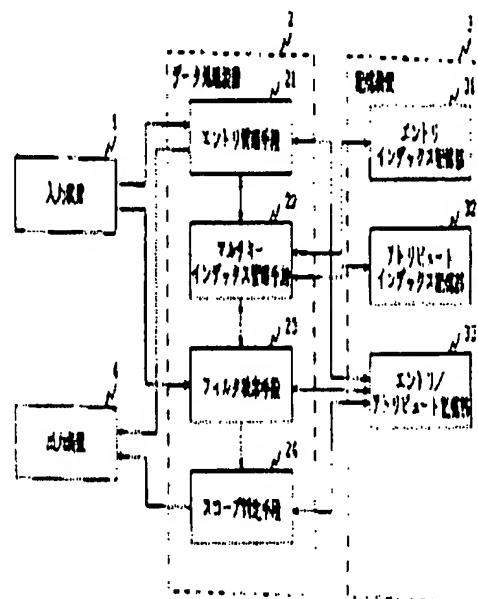
Fターム(参考) 5B082 EA01 EA05

(54) 【発明の名称】 ディレクトリ検索システム及び方法、ディレクトリ検索プログラムを記録したコンピュータ読取可能な記録媒体

(57) 【要約】

【課題】 ディレクトリ検索システムのインデックスを記憶する記憶容量を小さくし、インデックスの検索速度を上げる。

【解決手段】 アトリビュートインデックス記憶部32は、各エントリの複数のアトリビュートから構成されるマルチキーをキーとする2分探索木から構成されるマルチキーインデックスを記憶する。マルチキーインデックス管理手段22はマルチキーインデックスを管理し、フィルタ検索手段23は、フィルタ条件として複数のアトリビュートが指定されていても、マルチキーインデックス管理手段22を介してマルチキーインデックスを1回探索するだけでフィルタ条件を満たすエントリを検索することができる。



【特許請求の範囲】

【請求項1】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索システムであって、

前記各エントリの管理情報と前記各エントリの全ての属性とを前記ディレクトリ階層順に記憶するエントリ／アトリビュート記憶手段と、

エントリを識別するための識別子からエントリの前記ディレクトリ階層における階層番号および階層別番号を導き出すためのインデックスを記憶するエントリインデックス記憶手段と、

検索条件として指定された属性値を満たすエントリの識別子を取得するために前記各エントリの複数の属性から成るマルチキーをキーとする2分探索木から構成されるマルチキーインデックスを記憶するアトリビュートインデックス記憶手段と、

前記エントリインデックス記憶手段および前記アトリビュートインデックス記憶手段を管理するマルチキーインデックス管理手段と、

前記マルチキーインデックス管理手段を介して検索条件を満たすエントリの識別子を前記マルチキーインデックスから取得し、前記マルチキーインデックスから取得した識別子に基づいて検索条件を満たすエントリの階層番号および階層別番号を前記マルチキーインデックス管理手段を介して前記エントリインデックス記憶手段から取得し、該階層番号および階層別番号に基づいて検索条件を満たすエントリの管理情報を前記エントリインデックス記憶手段から取得するフィルタ検索手段とを備えるディレクトリ検索システム。

【請求項2】 前記エントリおよび前記属性の登録や削除や更新が要求された場合に、前記エントリ／アトリビュート記憶手段が記憶する前記エントリおよび前記属性の登録や削除や更新を行うとともに、前記マルチキーインデックス管理手段を介して前記エントリインデックス記憶手段に記憶されるインデックスと前記アトリビュートインデックス記憶手段に記憶されるマルチキーインデックスとに対しても登録や削除や更新を行うエントリ管理手段をさらに備える請求項1記載のディレクトリ検索システム。

【請求項3】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索システムであって、

前記各エントリの管理情報と前記各エントリの全ての属性とを前記ディレクトリ階層順に記憶するエントリ／アトリビュート記憶手段と、

エントリを識別するための識別子からエントリの前記ディ

レクトリ階層における階層番号および階層別番号を検索するためのインデックスを記憶するエントリインデックス記憶手段と、

検索条件として指定された属性値を満たすエントリの識別子を取得するために前記各エントリの複数の属性から成るマルチキーをキーとする2分探索木から構成される前記2分探索木の各ノードに格納されているエントリの前記階層番号および前記階層別番号が付与されているマルチキーインデックスを記憶するアトリビュートインデックス記憶手段と、

前記エントリインデックス記憶手段および前記アトリビュートインデックス記憶手段を管理するマルチキーインデックス管理手段と、

前記ディレクトリ階層における検索範囲が指定されていた場合に、検索条件として指定された属性の値に基づいて前記マルチキーインデックス管理手段を介して前記検索条件を満たすエントリを検索する途中に通過する前記各ノードに付与されている階層番号および階層別番号を有するエントリが前記検索範囲に入っているか否かを各エントリの先相関係からチェックし、該チェックの結果において前記検索範囲に含まれていなかったエントリを検索条件を満たすエントリから除外した後で、残りのエントリが前記検索範囲に含まれているか否かの判定を行うスコープ／フィルタ統合検索手段とを備えるディレクトリ検索システム。

【請求項4】 エントリの登録や削除や更新要求があった場合には前記エントリ／アトリビュート記憶手段が記憶するエントリの登録や削除や更新を行うとともに前記マルチキーインデックス管理手段を介して前記エントリインデックス記憶手段に記憶されるインデックスの登録や削除や更新を行い前記マルチキーインデックスの2分探索木の各ノードに付与されている階層番号および階層別番号についての更新も行うエントリ管理手段をさらに備える請求項3記載のディレクトリ検索システム。

【請求項5】 指定された文字列の中から所定の文字数の部分文字列をすべて抽出する部分文字列操作手段をさらに備え、

前記アトリビュートインデックス記憶部は前記部分文字列操作手段によって抽出された部分文字列をキーとするマルチキーインデックスを記憶しており、

前記マルチキーインデックス管理手段は、前記検索条件として文字列が含まれていた場合、前記部分文字列操作手段によって該文字列から抽出された少なくとも1つの部分文字列によって前記マルチキーインデックスの探索を行い、各部分文字列毎の前記マルチキーインデックスの探索結果となったエントリのうち、すべての部分文字列についての探索結果に含まれるエントリを検索条件を満たすエントリとする請求項1から4のいずれか1項記載のディレクトリ検索システム。

【請求項6】 属性となっている文字列の更新要求があ

った場合に、前記部分文字列操作手段から抽出された部分文字列のうち探索結果が同一になる部分文字列については、キー併合を行って該部分文字列が指すエントリの識別子を格納する場所を同一とする請求項5記載のディレクトリ検索システム。

【請求項7】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索方法であって、

前記各エントリの複数の属性から構成されたマルチキーをキーとする2分探索木から構成されるマルチキーインデックスを作成し、前記マルチキーインデックスから検索条件として指定された属性値を満たすエントリを検索するディレクトリ検索方法。

【請求項8】 前記エントリや前記属性の登録や削除や更新が行われる場合、前記エントリから構成されるディレクトリ階層および前記マルチキーインデックスに対しても登録や削除や更新を行う請求項7記載のディレクトリ検索方法。

【請求項9】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索方法であって、

エントリが有する複数の属性から成るマルチキーをキーとするエントリを特定するための2分探索木から構成され、前記各エントリを格納する前記2分探索木の各ノードに前記各エントリの前記ディレクトリ階層における階層番号および階層別番号がそれぞれ付与されているマルチキーインデックスを作成し、前記ディレクトリ階層における検索範囲が指定されていた場合に、検索条件として指定された属性の値に基づいて前記検索条件を満たすエントリを検索する途中に通過する前記各ノードに付与されている階層番号および階層別番号を有するエントリが前記検索範囲に入っているか否かを各エントリの先祖関係からチェックし、

該チェックの結果において前記検索範囲に含まれていなかったエントリを検索条件を満たすエントリから除外した後で、残りのエントリが前記検索範囲に含まれているか否かの判定を行うディレクトリ検索方法。

【請求項10】 エントリの登録や削除や更新が行われた場合には、ディレクトリ階層の登録や削除や更新を行うとともに、前記マルチキーインデックスの2分探索木の各ノードに付与されている階層番号および階層別番号の登録や削除や更新も行う請求項9記載のディレクトリ検索方法。

【請求項11】 前記マルチキーインデックスは、所定の文字数の部分文字列をキーとしており、検索条件として指定された属性値が文字列であった場

合、該文字列の中から所定の文字数の部分文字列をすべて抽出し、

検索条件として指定された属性値が文字列であった場合、該文字列から抽出された少なくとも1つの部分文字列によって前記マルチキーインデックスの探索を行い、各部分文字列毎の前記マルチキーインデックスの探索結果となったエントリのうち、すべての部分文字列についての探索結果に含まれるエントリを検索条件を満たすエントリとする請求項7から10のいずれか1項記載のディレクトリ検索方法。

【請求項12】 属性となっている文字列の更新要求があった場合に、前記部分文字列操作手段から抽出された部分文字列のうち探索結果が同一になる部分文字列については、キー併合を行って該部分文字列が指すエントリの識別子を格納する場所を同一とする請求項11記載のディレクトリ検索方法。

【請求項13】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、前記各エントリの複数の属性から構成されたマルチキーをキーとする2分探索木から構成されるマルチキーインデックスを用いてエントリを検索するディレクトリ検索システムに備えられ、

検索条件として指定された属性値を満たすエントリを前記マルチキーインデックスから検索する処理をコンピュータに実行させるためのプログラムを記録した機械読み取り可能な記録媒体。

【請求項14】 前記エントリや前記属性の登録や削除や更新が行われる場合、前記ディレクトリ階層および前記マルチキーインデックスに対しても登録や削除や更新を行う処理をコンピュータに実行させるためのプログラムをさらに記録した請求項13記載の機械読み取り可能な記録媒体。

【請求項15】 複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、前記各エントリが有する複数の属性から成るマルチキーをキーとするエントリを特定するための2分探索木から構成され前記各エントリを格納する前記2分探索木の各ノードに前記ディレクトリ階層における前記各エントリの階層番号および階層別番号がそれぞれ付与されているマルチキーインデックスを用いてエントリを検索するディレクトリ検索システムに備えられ、

前記ディレクトリ階層における検索範囲が指定されていた場合に検索条件として指定された属性の値に基づいて前記検索条件を満たすエントリを前記マルチキーインデックスを検索する途中に通過する前記各ノードに付与されている階層番号および階層別番号を有するエントリが前記検索範囲に入っているか否かを各エントリの先祖関係からチェックする処理と、

該チェックの結果において前記検索範囲に含まれていな

かったエントリを前記検索条件を満たすエントリから除外した後で、残りのエントリが前記検索範囲に含まれているか否かの判定を行う処理とをコンピュータに実行させるためのプログラムを記録した機械読み取り可能な記録媒体。

【請求項16】 利用者からエントリの登録や削除や更新が行われた場合には、前記ディレクトリ階層の登録や削除や更新を行うとともに前記2分探索木の各ノードに付与されている階層番号および階層別番号の登録や削除や更新も行う処理をコンピュータに実行させるためのプログラムをさらに記録した請求項15記載の機械読み取り可能な記録媒体。

【請求項17】 検索条件として指定された属性値が文字列であった場合、該文字列の中から所定の文字数の部分文字列をすべて抽出する処理と、

検索条件として指定された属性値が文字列であった場合に該文字列から抽出された少なくとも1つの部分文字列によって前記所定の文字数の部分文字列をキーの1つとするマルチキーインデックスの探索を行い、前記各部分文字列毎の前記マルチキーインデックスの探索結果となったエントリのうちすべての部分文字列についての探索結果に含まれるエントリを前記検索条件を満たすエントリとする処理とをコンピュータに実行させるためのプログラムをさらに記録した請求項13から16のいずれか1項記載の機械読み取り可能な記録媒体。

【請求項18】 利用者から属性となっている文字列の更新要求があった場合に前記文字列から抽出された部分文字列のうちの探索結果が同一になる部分文字列についてはキー併合を行って該部分文字列が指すエントリの識別子を格納する場所を同一とする処理をコンピュータに実行させるためのプログラムをさらに記録した請求項17記載の機械読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データベースなどの検索を行うディレクトリ検索システムおよび方法、ディレクトリ検索用プログラムを記録したコンピュータ読み取り可能な記録媒体に関し、特に、ディレクトリサービスにおいてエントリを検索するディレクトリ検索システムおよび方法、インデックス用プログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】 ディレクトリ検索システムは、ディレクトリ階層上に管理されるデータのデータベースの検索を高速に行うために用いられるシステムである。ディレクトリ検索システムは、2分探索木やハッシュ表のような仕組みを保持している。この仕組みをインデックスという。インデックスは、データを特定するためのキーに基づいて構成されている。ディレクトリ検索システムは、検索条件として指定されたキーの値によってインデック

スを探索してデータを特定する。

【0003】 特開昭60-254325号公報にはディレクトリ検索システムの一例が開示されている。その公報に開示されているディレクトリ検索システムは、インデックスを主記憶装置上に保持している。このディレクトリ検索システムは、データを特定するキーによってインデックスを検索してそのデータの位置情報を取得し、2次記憶装置上の目的のデータにアクセスする。一般的に、主記憶装置は2次記憶装置よりもはるかにアクセス速度が速い。そのため、このようなディレクトリ検索システムを用いることによって、2次記憶装置（外部記憶装置）上に大量のデータが格納されていても目的のデータに高速にアクセスすることができる。通常、このようなインデックスは1種類のキーに基づいて構成されている。また、ディレクトリ検索システムでは1種類のキーに対して複数のインデックスを検索しなければならない場合もある。この場合、1種類のキーについて複数のインデックスを検索するため、検索に時間がかかってしまうという問題があった。特開平3-70049号公報には、この問題を解決するディレクトリ検索システムが開示されている。図15は、この公報で開示されたディレクトリ検索システムの構成を示すブロック図である。

【0004】 このディレクトリ検索システムは、レコード処理の開始を指示する処理開始指示装置1001とディレクトリ情報処理手段1002と併合インデックス作成手段1005と併合インデックス読込手段1006と作業ファイル1007とディレクトリファイル1009とを備えている。

【0005】 ディレクトリ情報処理手段1002は、併合インデックス作成指示手段1003と併合インデックス読込指示手段1004とを備えている。ディレクトリファイル1009は、データベースのデータの管理情報を記憶するためのものである。ディレクトリファイル1009は、データの管理情報の部分集合である幾つかのサブファイルから成るサブファイル群1013と、利用者インデックス1010と、グループ共有インデックス1011と、システム共有インデックス1012とを備えている。利用者インデックス1010は利用者毎に作成されており、ディレクトリファイル内に多数記憶されている管理情報レコードのうち、その利用者のみが利用できるデータの管理情報レコードを管理している。グループ共有インデックス1011は複数の利用者から構成されるグループ毎に作成されており、グループに属する利用者のみが利用できるデータの管理情報レコードを管理している。システム共有インデックス1012は1つだけ作成されており、全利用者が利用できるデータの管理情報レコードを管理している。

【0006】 このディレクトリ検索システムは以下のように動作する。まず、処理開始指示装置1001からディレクトリ情報処理手段1002に対し処理開始が要求

されると、併合インデックス作成指示手段1003は、入力された処理指示者名に基づいて併合インデックス作成手段1005に対し併合インデックスを作成するように指示する。

【0007】すると、併合インデックス作成手段1005は、ディレクトリファイル1009内の処理指示者の利用者インデックス1010と処理指示者が属するグループ共有インデックス1011とシステム共有インデックス1012とを読み込み、これらを併合して作業ファイル1007上に併合インデックス1008を作成する。次に、併合インデックス読込指示手段1004は、併合インデックス読込手段1006に対し併合インデックス1008を読込むように指示する。併合インデックス読込手段1006は管理情報レコードを特定するキーである管理情報レコード名を用いて併合インデックス1008から該当する管理情報レコードの位置情報を読み、それをディレクトリ情報処理手段1002へ渡す。ディレクトリ情報処理手段1002は渡された管理情報レコードの位置情報により目的の管理情報レコードをサブファイルから読込む。

【0008】このディレクトリ検索システムでは、データを探索するためのキーとなる管理情報レコード名から目標とする管理情報レコードを読み込むのに、3つのインデックスを別々に探索しなければならなかったところを、3つのインデックスを併合し、併合されたインデックスを1回探索するだけでインデックスレコードを求めることができるので、検索時間を短縮することができる。このように、ディレクトリ検索システムでは、インデックスを探索する時間を短縮することが1つの課題となっている。

【0009】また、ディレクトリ検索システムでは、データを探索するときに数種類のキーを用いる場合が多い。この場合、従来のディレクトリ検索システムでは、それぞれのキーに対応したインデックスを1つずつ探索しなければならないため、データの検索時間が長くなってしまいう問題があった。

【0010】数種類のキーから一度にデータを探索することができる2分探索木がいくつか提案されている。この2分探索木の1つにキーデー・ツリー（以降 k-d tree）がある。k-d treeは、アイ・イー・イー・イー・トランザクションズ・オン・ソフトウェア・エンジニアリング、第SE-5巻、第4号、333~340頁（IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-5, NO. 4, JULY 1979, Pages 333-340.）に掲載されたマルチディメンショナル・バイナリ・サーチ・ツリーズ・イン・データベース・アプリケーションズ（Multidimensional Binary Search Trees in Database Application

s）と題するジョン・L・ベントレー（John L. Bentley）による論文に記載されている。

【0011】このk-d treeは、2分探索木の各ノードに検索するデータのエンTRIESが有する複数のキー値を格納できるように拡張されたものである。k-d treeでは、ツリーの格段毎に各キーが順番に分岐条件として用いられる。例えば、キーがその人の名前、年齢、電話番号であるとする、第1段目では名前が分岐条件として用いられ、第2段目では年齢が分岐条件として用いられ、第3段目は電話番号が分岐条件として用いられ、第4段目以降も名前、年齢、電話番号がこの順番で分岐条件として用いられる。

【0012】図16は、k-d Treeの基本的な構成を示す図である。このk-d treeは、ある会社の社員を探索するためのツリーである。このツリーでは、社員の名前と年齢とが、社員を探索するためのキーとなる。

【0013】図16に示すように、根（以降 ルート）のノード101には、Sato氏（Sato, 22）が格納されている。Satoは名前を表し、22は年齢を表す。このツリーの第1段目では、名前が分岐する。ルートにおける名前の頭文字はSであるので、名前の頭文字のアルファベットがA~Rである人は左へ分岐し、名前の頭文字のアルファベットがS~Zである人は右へ分岐する。したがって、Endo氏（Endo, 28）は左へ分岐してノード102に格納され、Ueda氏（Ueda, 34）は右へ分岐してノード103に格納される。ツリーの第2段目では年齢によって分岐する。年齢が分岐元のノードに格納されている年齢より下の人は左に分岐し、年齢が分岐元のノードに格納されている年齢以上の人は右に分岐する。したがって、Iijima氏（Iijima, 20）は左へ分岐してノード104に格納され、Aoki氏（Aoki, 32）は右に分岐してノード105に格納される。第3段目では、第1段目と同様に名前のアルファベットで分岐する。したがって、Doi氏（Doi, 22）は左へ分岐してノード106に格納され、Kato氏（Kato, 25）は右へ分岐してノード107に格納される。そして、Nakai氏（Nakai, 37）はノード105から右に分岐してノード108に格納される。

【0014】k-d treeでは、指定されていないキーによって分岐されている段において枝の選択は行われない。例えば、年齢が検索条件として指定されていない場合には、2段目ではノード104およびノード105の両方へ進むようにする。以上述べたように、k-d treeでは、数種類のキーを用いて一度にデータの検索が行えるので、検索時間を短縮することができる。

【0015】しかし、このようなk-d treeでは、木のデータの分布が不均衡である場合に検索条件によって検索時間に偏りがあるという問題があった。k-

d Freeを改良して、木のデータ分布が不均衡な場合でも検索時間が安定する2分探索木として、エッチ・ビー・ツリー（以降 hB tree）がある。hB treeは、1990年12月、エー・シー・エム・トラザクションズ・オン・データベース・システムズ、第15巻、第4号、624～658頁（ACM Transactions on Database Systems, Vol. 15, No. 4, December 1990, Pages 625-658.）に掲載されたザ・エッチ・ビー・ツリー：マルチディメンショナル・インデックス・メソッド・ウィズ・グッド・ギャランティード・パフォーマンス（The hB Tree: A Multiattribute Indexing Method with Good Guaranteed Performance）と題するデビッド・B. ロメット（David B. Lomet）らによる論文に記載されている。

【0016】k-d treeは、オブジェクト単位でインデックスを管理しているが、hB treeは、複数のオブジェクトを含む所定の範囲でインデックスを管理している。この所定の範囲をブリックという。各ブリックには、前述のk-d tree方式のインデックスが格納されている。ブリック内におけるオブジェクトの数が所定の値を超えた場合、そのブリックは分割される。分割されたブリック同士は2分探索木方式で接続され、その木におけるノードとなる。そのブリックにより構成された2分探索木のリーフ部分には、各オブジェクトの情報が格納される。このhB treeで目的のオブジェクトを検索するときには、まず、検索条件を満たすブリックが探索され、そのブリック内のk-d treeを検索して目的のオブジェクトの情報が格納されるリーフが探索される。

【0017】図17は、図16のk-d treeで示された社員の分布状態を示すマップである。このマップは、縦軸が年齢であり、横軸が名前となっている。図17に示すように、アルファベットSをx3とし、アルファベットIをx2とし、年齢28をy1とする。マップ上のエリアは、年齢28以上かつ名前の頭文字がAの領域301と、年齢28以上で名前の頭文字がB～Rの領域302と、年齢28以下で名前の頭文字がA～Hの領域303と、年齢28以下で名前の頭文字がI～Rの領域304と、名前の頭文字がS～Zの領域305とに分割されている。図18は、これらの社員の構成をhB treeで表した図である。このhB treeは、ノード201～203と、リーフ204～208とから構成されている。ノード201～203は、それぞれhB treeのブリックを表すものであり、それぞれをブリックa、ブリックb、ブリックcとする。図18のhB treeでは、図16におけるマップの各領域を元にブリックa、b、cを構成する。ブリックaは、マップ全

体を指すものであるとし、ブリックbは、領域301、302、305を指すものであるとし、ブリックcは、領域303、304を指すものであるとする。

【0018】領域303、304には、Iijima氏、Doi氏、Kato氏が含まれている。この3氏は、図16のk-d treeにおけるノード104以下に格納されている。したがって、ブリックbには、図16のk-d treeのノード104以下の部分がk-d tree方式で管理されている。

【0019】また、領域301、302、305には、Sato氏、Endo氏、Ueda氏、Aoki氏、Nakai氏が含まれている。この5人は、図16のk-d treeにおけるノード101、102、103、105、108に格納されている。したがって、ブリックcには、それらのノード101、102、103、105、108から構成される部分がk-d tree方式で管理されている。ブリックaは、ブリックbへ分岐するかブリックcへ分岐するかを決定するためのものである。図16のk-d treeによれば、ブリックbへ分岐するかブリックcへ分岐するかは、ノード101またはノード102によって決定されるため、ブリックa内のk-d treeは、ノード101とノード102から構成され、そのk-d treeの葉には、選択すべきブリック名が指定されている。

【0020】リーフ204～リーフ208には、E、D、C、B、Aのリーフ名がそれぞれに付与されており、図17に示す領域301～305に含まれる社員についての情報が格納されている。リーフ204には領域303に含まれるDoi氏（Doi, 22）に関する情報が格納され、リーフ204には領域304に含まれるIijima氏（Iijima, 20）、Kato氏（Kato, 22）に関する情報が格納される。以下リーフ207、208には領域302、305に含まれる社員に関する情報が格納される。

【0021】また、ブリックb、cは前述のとおり、k-d tree方式により構成されるが、そのツリーの葉には、そのリーフ204～208のうち選択すべきリーフ名が指定されている。また、該当するオブジェクトがない場合は、その葉は外部ノードであるとしてextが指定される。各ブリックの各ノードにあるx1、x2、x3、y1は、それぞれ図18に示すように、アルファベットのA、アルファベットのI、アルファベットのS、年齢の28を示し、各ノードにおける分岐条件を示す。

【0022】以上述べたように、データを検索する2分探索木をhB treeとすることによって、複数のオブジェクトを管理するブリックを2分探索木のノードとして木の枝の状態を均一化することができるため、安定した検索時間を得ることができる。なお、上述のような複数のキーから構成されたマルチキーをキーとしてデー

タを特定するための2分探索木から構成されるインデックスをマルチキーインデックスという。

【0023】一方、特開平10-289139号公報に開示されているようなディレクトリサーバでは、各ファイルの媒体情報を一元管理して保存しておき、クライアントからの検索要求や更新要求があった場合、一元管理してある媒体情報の検索や更新などを行う。この媒体情報の管理などにも、上述のようなディレクトリ検索システムが用いられている。

【0024】また、媒体情報の管理を始めとして、個人情報管理やネットワーク管理などに利用される汎用のディレクトリサービスでは、組織や個人、コンピュータ、プリンタなど、様々な資源が管理対象となる。ディレクトリ検索システムは、それらの資源をエントリとして、資源間の関係に基づいてディレクトリ階層を形成し、それらのエントリの属性（以降 アトリビュート）をキーとしたインデックスを探索することによって、それらのエントリを検索することができる仕組みを提供している。

【0025】このようなディレクトリ検索システムの一例が特願平11-43259号出願で示されている。このディレクトリ検索システムは、エントリが有するアトリビュートについての検索条件であるフィルタ条件とディレクトリ階層に基づいて設定される検索範囲であるスコープ条件とを利用者が指定することによってエントリを検索するシステムである。

【0026】図19は、この出願で示されているディレクトリ検索システムの構成を示すブロック図である。このディレクトリ検索システムは、入力装置1と、データ処理装置8と、記憶装置9と、出力装置4とから構成される。入力装置1は、キーボード等の利用者からの要求を入力するためのものである。データ処理装置8はプログラム制御により動作し、記憶装置9が記憶する情報の検索を行う。出力装置4はディスプレイ装置や印刷装置等であり、データ処理装置8が行った処理の結果を出力する。記憶装置9は、エントリ／アトリビュート記憶部31と、複合インデックス記憶部35と、先祖関係記憶部34とを備えている。

【0027】エントリ／アトリビュート記憶部31は、全てのエントリの情報と、その各エントリが有する属性であるアトリビュートの情報を記憶する。

【0028】複合インデックス記憶部35は、各エントリがエントリ／アトリビュート記憶部31におけるディレクトリ上の何階層目の何番目にあるかという複合したデータを、2分探索木により構成されるインデックスやハッシュ表等の形式で記憶している。

【0029】図20は、ある会社の組織をディレクトリ階層で表す図である。このディレクトリ階層は4層から構成される。階層1は会社、階層2は部門、階層3は課、階層4は社員の層となっている。会社401には、

営業部門402と開発部門403とがある。営業部門402には国内営業課404と海外営業課405、開発部門403には企画課406と製造課407という課がある。各課にはそれぞれの人408～415が配属されている。

【0030】先祖関係記憶部34は、各エントリの先祖関係を記憶している。図21は、先祖関係記憶部34に記憶されている先祖関係を表す先祖関係表を示す図である。この表は、各行にはエントリ／アトリビュート記憶部31が記憶する各エントリのうち、先祖を有するエントリが各ディレクトリ階層毎に並べられている。また、この表の各列には、エントリ／アトリビュート記憶部31が記憶する各エントリのうち、子孫を有するエントリが各ディレクトリ階層毎に並べられている。この表上でチェックされているところは、列のエントリが、行のエントリの先祖になっていることを示している。例えば、Ueda氏の先祖は、会社401と、営業部門402と、海外営業課405であるということが、この表からわかるようになっている。

【0031】データ処理装置8は、エントリ管理手段21と、フィルタ検索手段27と、スコープ判定手段28とから構成される。エントリ管理手段21は、エントリやアトリビュートの登録・削除・更新の処理を行う。フィルタ検索手段27は、利用者が指定した検索条件にしたがってエントリの検索を行う。スコープ判定手段28は、フィルタ検索手段27が検索したエントリが利用者が指定した検索範囲に入っているかどうか調べてエントリを絞り込む。

【0032】まず、利用者からの要求が、エントリやアトリビュートの更新要求、削除要求、登録要求のいずれかの要求であった場合のディレクトリ検索システムの動作について説明する。入力装置1からその要求が入力されると、エントリ管理手段21は、その要求がアトリビュート更新要求ならばエントリ／アトリビュート記憶部31に格納されているエントリのアトリビュートを更新する。

【0033】また、その要求がID更新又は削除要求ならば、エントリ管理手段21は、複合インデックス記憶部35に記憶されているエントリのIDを更新または削除し、先祖関係記憶部34の先祖関係表におけるそのエントリに関係する箇所のチェックも更新または削除する。また、その要求がエントリまたはアトリビュートの削除要求ならば、エントリ管理手段21は、エントリ／アトリビュート記憶部31に格納されたデータの中から該当するエントリ又はアトリビュートを削除する。また、その要求がエントリまたはアトリビュートの登録要求ならば、エントリ管理手段21は、エントリ／アトリビュート記憶部31に登録要求されたエントリまたはアトリビュートを登録する。

【0034】その要求がID更新又は登録要求の場合、

エントリ管理手段21は、複合インデックス記憶部35にエントリのIDを追加し、先祖関係記憶部34の先祖関係表の該当箇所にもチェックする。上述の処理が終了したら、エントリ管理手段21は処理終了を出力装置4に通知し、出力装置4は処理結果を表示する。

【0035】次にアトリビュートをキーとして、エントリを検索する場合の動作について説明する。入力装置1から与えられた検索要求がフィルタ検索手段27に入力されると、フィルタ検索手段27は、指定された検索条件であるフィルタ条件にしたがってエントリ／アトリビュート記憶部31に格納されているエントリの検索を行う。検索条件を満たしたエントリは、フィルタ検索手段27によって一時集合（不図示）に格納される。その後、スコープ判定手段28は、一時集合に格納されたエントリのうち検索範囲に入っているエントリを先祖関係記憶部34の先祖関係表を参照することによって抽出し、結果集合（不図示）に格納する。その後、フィルタ検索手段27は、結果集合に格納されたエントリについて複合インデックス記憶部35にアクセスしてそのエントリのディレクトリ情報を取得し、その検索結果を出力装置4に出力する。

【0036】以上述べたように、このディレクトリ検索システムでは、フィルタ検索手段27によってエントリ／アトリビュート記憶部31よりエントリを検索した後、その検索されたエントリの中から、スコープ判定手段28によって、検索範囲に入っているエントリを抽出している。よって、フィルタ検索手段27において検索されたエントリの中には検索範囲外にあるエントリも含まれている場合があり、それらのエントリの検索が結果的に無駄となり、検索時間が長くなってしまいうという問題点があった。

【0037】また、このようなアトリビュートによるエントリのフィルタ検索では、エントリのアトリビュートに文字列や数値が含まれていた場合、文字列や数値の全一致検索だけではなく、部分一致検索もすることができるディレクトリ検索システムが要求される。

【0038】また、特開平10-187745号公報では、エントリのアトリビュートに文字列や数値がフィルタ条件として含まれている場合に、その文字列の部分一致検索を行うことができるディレクトリ検索システムが開示されている。しかしながらこの公報に開示された従来のディレクトリ検索システムでは、アトリビュートの全一致検索用のインデックスと、部分一致検索用のインデックスを別々に備えることが必要となる。さらに、部分一致検索用のインデックスでは、格納されるアトリビュートが膨大な数にのぼるため、インデックスの大きさも大きくなる。そのため、文字列の部分一致検索が可能な従来のディレクトリ検索システムでは、それらのインデックスをできるだけ統合して、インデックスに使用される記憶装置の容量をできるだけ抑制することが課題と

なっている。

【0039】

【発明が解決しようとする課題】上述のような従来のディレクトリ検索システムは以下に示す3つの問題点を有している。

(1) 数種類のキーからデータを検索する場合には複数のインデックスを必要とするため、複数のインデックスを主記憶装置または2次記憶装置上に展開させなければならないので大容量の記憶装置が必要である。また、複数のインデックスを検索するので検索時間が長くなってしまふ。

(2) ディレクトリサービスにおいてディレクトリ検索システムを用いてエントリを検索する場合、フィルタ条件に基づいてエントリの検索を行うフィルタ処理と、検索範囲に基づいてエントリの絞り込みを行うスコープ処理とが別々に実行される。そのため、フィルタ処理において検索されたエントリが、検索範囲外である場合があり、フィルタ処理において行われた検索処理が一部無駄となり、検索時間が長くなってしまふ。

(3) アトリビュートの部分一致の検索が可能な従来のディレクトリ検索システムでは、アトリビュートの全一致検索用のインデックスと、部分一致検索用のインデックスという複数のインデックスが必要である。また、部分一致検索用のインデックスは、格納されるアトリビュートが膨大な数にのぼるため、インデックスのサイズも大きくなる。よって、アトリビュートの部分一致の検索が可能な従来のディレクトリ検索システムでは、大容量の記憶装置が必要となってしまふ。

【0040】よって、本発明は、キーが2種類以上指定された場合でも、インデックスを記憶するのに必要な記憶容量を少なくし、検索時間を短縮することができるディレクトリ検索システムを提供することを目的とする。

【0041】また、本発明は、検索条件および検索範囲の両方について検索を行わなければならない場合でも、検索時間を短縮することができるディレクトリ検索システムを提供することを目的とする。

【0042】また、本発明は、エントリのアトリビュートに文字列が含まれていて、その文字列の部分一致検索を行う場合でも、できるだけインデックスを記憶するのに必要な記憶容量を少なくするディレクトリ検索システムを提供することを目的とする。

【0043】

【課題を解決するための手段】上記問題点を解決するために、本発明では、複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索システムであって、前記各エントリの管理情報と前記各エントリの全ての属性とを前記ディレクトリ階層順に記憶するエントリ／アトリビュート記憶手段と、エン

トリを識別するための識別子からエントリの前記ディレクトリ階層における階層番号および階層別番号を導き出すためのインデックスを記憶するエントリインデックス記憶手段と、検索条件として指定された属性値を満たすエントリの識別子を取得するために前記各エントリの複数の属性から成るマルチキーをキーとする2分探索木から構成されるマルチキーインデックスを記憶するアトリビュートインデックス記憶手段と、前記エントリインデックス記憶手段および前記アトリビュートインデックス記憶手段を管理するマルチキーインデックス管理手段と、前記マルチキーインデックス管理手段を介して検索条件を満たすエントリの識別子を前記マルチキーインデックスから取得し前記マルチキーインデックスから取得した識別子に基づいて検索条件を満たすエントリの階層番号および階層別番号を前記マルチキーインデックス管理手段を介して前記エントリインデックス記憶手段から取得し該階層番号および階層別番号に基づいて検索条件を満たすエントリの管理情報を前記エントリインデックス記憶手段から取得するフィルタ検索手段とを備える。

【0044】本発明のディレクトリ検索システムでは、アトリビュートインデックス記憶手段が記憶するインデックスが、複数のキーによって一度に探索可能なマルチキーインデックスとなっているので、複数のインデックスを備える必要がないため、アトリビュートからエントリを検索するためのインデックスに必要な記憶容量を少なくことができる。また、複数のインデックスを検索する必要がないので検索時間を短縮することができる。

【0045】本発明の他のディレクトリ検索システムは、複数の資源を管理するために該各資源をエントリとして該各資源間の関係に基づいてディレクトリ階層を構成し、該各エントリの属性をキーとするインデックスを用いてエントリを検索するディレクトリ検索システムであって、前記各エントリの管理情報と前記各エントリの全ての属性とを前記ディレクトリ階層順に記憶するエントリ／アトリビュート記憶手段と、エントリを識別するための識別子からエントリの前記ディレクトリ階層における階層番号および階層別番号を検索するためのインデックスを記憶するエントリインデックス記憶手段と、検索条件として指定された属性値を満たすエントリの識別子を取得するために前記各エントリの複数の属性から成るマルチキーをキーとする2分探索木から構成され前記2分探索木の各ノードに格納されているエントリの前記階層番号および前記階層別番号が付与されているマルチキーインデックスを記憶するアトリビュートインデックス記憶手段と、前記エントリインデックス記憶手段および前記アトリビュートインデックス記憶手段を管理するマルチキーインデックス管理手段と、前記ディレクトリ階層における検索範囲が指定されていた場合に、検索条件として指定された属性の値に基づいて前記マルチキーインデックス管理手段を介して前記検索条件を満たすエ

ントリを検索する途中に通過する前記各ノードに付与されている階層番号および階層別番号を有するエントリが前記検索範囲に入っているか否かを各エントリの先祖関係からチェックし、該チェックの結果において前記検索範囲に含まれていなかったエントリを検索条件を満たすエントリから除外した後で、残りのエントリが前記検索範囲に含まれているか否かの判定を行うスコープ／フィルタ統合検索手段とを備える。

【0046】本発明のディレクトリ検索システムでは、フィルタ検索とスコープ判定の処理を統合するスコープ／フィルタ統合検索手段を備えることにより、フィルタ検索に含まれていてスコープ判定に含まれないエントリにアクセスすることがなくなるため、エントリの検索時間を短縮することができる。

【0047】本発明のディレクトリ検索システムでは、マルチキーインデックスの探索時にスコープの絞り込みを行い、スコープの判定処理の負荷を軽減しているため、スコープ判定処理の負荷を軽減してエントリの検索時間を短縮することができる。本発明の他のディレクトリ検索システムは、指定された文字列の中から所定の文字数の部分文字列をすべて抽出する部分文字列操作手段をさらに備え、前記アトリビュートインデックス記憶部は前記部分文字列操作手段によって抽出された部分文字列をキーとするマルチキーインデックスを記憶しており、前記マルチキーインデックス管理手段は、前記検索条件として文字列が含まれていた場合、前記部分文字列操作手段によって該文字列から抽出された少なくとも1つの部分文字列によって前記マルチキーインデックスの探索を行い、各部分文字列毎の前記マルチキーインデックスの探索結果となったエントリのうち、すべての部分文字列についての探索結果に含まれるエントリを検索条件を満たすエントリとする。

【0048】本発明のディレクトリ検索システムでは、文字列から所定の文字数の部分文字列を抽出する部分文字列操作手段と、すべての部分文字列の検索結果に含まれるエントリを検索結果であると判断するマルチキーインデックス管理手段とを備えることによって、同一のインデックスで全文字列および部分文字列の検索を行うことができるため、インデックスに必要な記憶容量を少なくすることができる。

【0049】

【発明の実施の形態】次に、本発明の実施形態のディレクトリ検索システムについて図面を参照して詳細に説明する。全図において、同一の符号がつけられている構成要素は、すべて同一のものを示す。

【0050】（第1の実施形態）まず、本発明の第1の実施形態のディレクトリ検索システムについて説明する。本実施形態のディレクトリ検索システムは、図20の会社401の組織を管理するデータベースに用いられているものとする。図1は、本実施形態のディレクトリ

検索システムの構成を示すブロック図である。図1に示すように、本実施形態のディレクトリ検索システムは、入力装置1とデータ処理装置2と記憶装置3と出力装置4とから構成されている。

【0051】入力装置1は、キーボード等であり、利用者からの要求を入力し、その要求をデータ処理装置2に出力している。出力装置4は、ディスプレイ装置や印刷装置等の出力装置であり、データ処理装置3から処理の終了通知を受け取って処理結果を表示する。データ処理装置2は、エントリ管理手段21と、マルチキーインデックス管理手段22と、フィルタ検索手段23と、スコープ判定手段24とを備えている。また、記憶装置3は、エントリインデックス記憶部31とアトリビュートインデックス記憶部32とエントリ/アトリビュート記憶部33とを備えている。

【0052】エントリ/アトリビュート記憶部33は、全てのエントリの管理情報と各エントリが有する全てのアトリビュートとを記憶している。各エントリは、ディレクトリ階層の形態で記憶されており、ディレクトリのルートであるベースエントリの管理情報の格納場所と各エントリの階層番号および階層別番号が分かれば、各エントリの管理情報の格納場所を特定でき、各エントリの管理情報にアクセスすることができる。本実施形態のディレクトリ検索システムでは、図20のディレクトリ階層の各エントリの管理情報と各エントリが有する全てのアトリビュートとを記憶している。

【0053】エントリインデックス記憶部31は、エントリを識別するための識別子（以降DN: Distinguished Name）に基づいて作成されたインデックスを記憶している。このインデックスを検索することによって各エントリの階層番号と階層別番号とを取得することができる。

【0054】アトリビュートインデックス記憶部32は、各エントリが有するアトリビュートをキーとして構成されたインデックスを記憶している。このインデックスは、前述のh B t r e e方式を用いたマルチキーインデックスとなっており、複数のアトリビュートをキーとしてエントリを検索することができるようになってい。本実施形態のディレクトリ検索システムでは、アトリビュートインデックス記憶部32に記憶されるマルチキーインデックスは、図18のようになる。

【0055】エントリ管理手段21は、入力装置1からエントリの登録・削除・ID更新要求やアトリビュートの登録・削除・更新要求を入力した場合、エントリ/アトリビュート記憶部33に対してエントリやアトリビュートの登録・削除・更新を行う。さらに、エントリ管理手段21は、マルチキーインデックス管理手段22を介して、エントリインデックス記憶部31およびアトリビュートインデックス記憶部32の登録、削除、更新を行う。

【0056】フィルタ検索手段23は、入力装置1からエントリの検索要求を入力した場合、指定されたフィルタ条件を満たすエントリの検索を行う。スコープ判定手段24は、フィルタ検索手段23によって検索されたエントリが、指定された検索範囲であるスコープ条件を満たすかどうかを判定する。スコープ判定手段24は、エントリ/アトリビュート記憶部33のエントリのディレクトリ階層を参照することによってその判定を行う。

【0057】マルチキーインデックス管理手段22は、フィルタ検索手段23やエントリ管理手段21からの要求により、エントリインデックス記憶部31やアトリビュートインデックス記憶部32が記憶するインデックスの管理や探索を行う。

【0058】次に、本実施形態のディレクトリ検索システムの動作について図2、図3、図18を参照して詳細に説明する。図2は、エントリやアトリビュートの更新要求が入力された場合の本実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【0059】利用者が入力装置1に対し、エントリやアトリビュートの更新要求を入力した場合、入力装置1はその要求をエントリ管理手段21へ出力する。すると、エントリ管理手段21は、その要求がエントリの更新要求であるか否かを判定する（ステップA1）。もし、その要求がエントリの更新要求であれば、エントリ管理手段21は、マルチキーインデックス管理手段22を介してエントリインデックス記憶部31が記憶するインデックスを更新する（ステップA2）。その後、エントリ管理手段21は、エントリ/アトリビュート記憶部33に記憶されているエントリの更新を行う（ステップA3）。ステップA2、A3の処理終了後、エントリ管理手段21は処理終了を出力装置4に通知し、出力装置4は処理結果を出力する（ステップA4）。

【0060】ステップA1において、入力装置1からの要求がエントリの更新要求ではない場合、その要求はアトリビュートの更新要求である。エントリ管理手段21は、アトリビュートインデックス記憶部32において更新しようとするアトリビュートにインデックスが付与されているか否かを判定する（ステップA5）。更新しようとするアトリビュートにインデックスが付与されていれば、エントリ管理手段21は、マルチキーインデックス管理手段22を介してアトリビュートインデックス記憶部32に記憶されるアトリビュートのインデックスを更新する（ステップA6）。エントリ管理手段21は、エントリ/アトリビュート記憶部33に記憶されるアトリビュートを更新する（ステップA7）。そして、ステップA6、A7の処理終了後、エントリ管理手段21は処理終了を出力装置4に通知し、出力装置4は処理結果を表示する（ステップA4）。

【0061】ステップA5において、更新しようとするアトリビュートにインデックスが付与されていない場

合、ステップA6の処理は実行されずに、ステップA7において、エントリ／アトリビュート記憶部33のアトリビュートの更新のみが行われる。

【0062】ステップA6において、アトリビュートのインデックスを更新する際の動作について説明する。前述のように、アトリビュートインデックス記憶部32は、図18に示す社員の名前と年齢とをキーとしたマルチキーインデックスを記憶している。なお、更新されるアトリビュートは、社員の名前と年齢であり、更新されるアトリビュート値は(Ozawa, 24)であるとする。

【0063】このマルチキーインデックスでは、ブリックaから探索が開始される。ブリックaでは、「O」とx3(S)とが比較され、「24」とy1(28)とが比較されることによって、ブリックbが選択される。ブリックbでは、「O」とx2(1)とが比較され、リーフ205が選択される。リーフ205に(Ozawa, 24)のアトリビュート値を有するエントリのDNが格納される。

【0064】図3は、エントリを検索する場合の本実施形態のディレクトリ検索システムの動作を示すフローチャートである。利用者が入力装置1に対しエントリの検索要求を入力した場合、入力装置1はその要求をフィルタ検索手段23へ出力する。フィルタ検索手段23は、エントリの検索要求が入力されるとマルチキーインデックス管理手段22を介してエントリインデックス記憶部31のインデックスを探索する(ステップB1)。そのインデックスのルートであるベースエントリのエントリ／アトリビュート記憶部33における格納場所を取得する(ステップB2)。そして、フィルタ検索手段23はフィルタ条件が含まれているか否かを調べる(ステップB3)。もし、検索要求にフィルタ条件が含まれていれば、フィルタ検索手段23は、フィルタ条件となっているアトリビュートにインデックスが付与されているかを調べる(ステップB6)。

【0065】フィルタ条件となっているアトリビュートにインデックスが付与されていれば、フィルタ検索手段23は、マルチキーインデックス管理手段22を介してそのフィルタ条件のアトリビュートの値によってマルチキーインデックスの探索を行い、フィルタ条件を満たすエントリのDNを得る。(ステップB7)。そして、フィルタ検索手段23は、マルチキーインデックス管理手段22を介してエントリインデックス記憶部31が記憶するインデックスを探索してステップB7において検索されたエントリのDNからそのエントリの階層番号と階層別番号を得る(ステップB8)。フィルタ検索手段23は、エントリ／アトリビュート記憶部33からステップB8において得られたベースエントリの格納場所とエントリの階層番号および階層別番号とからエントリの格納場所を割り出し、そのエントリの情報を取得する(ス

テップB9)。

【0066】ステップB6において、フィルタ条件となっているアトリビュートにインデックスが付与されていない場合には、フィルタ検索手段23は、データベースの検索機能などを利用して、エントリ／アトリビュート記憶部33からフィルタ条件を満たすエントリの検索を行って(ステップB10)、ステップ9に移行する。ステップB9の後およびステップB3において検索要求にフィルタ条件が含まれていなかったときは、本実施形態のディレクトリ検索システムの動作はスコープ判定手段24に移行する。スコープ判定手段24は、検索要求にスコープ条件が含まれているかを調べる(ステップB4)。スコープ判定手段24は、入力装置1から入力された検索要求にスコープ条件が含まれていれば、エントリ／アトリビュート記憶部33のディレクトリ階層をたどってステップB9で取得されたエントリが、スコープ条件を満たしているか否かの判定を行う(ステップB11)。そして、スコープ判定手段24は、スコープ条件を満たしているエントリをエントリ／アトリビュート記憶部33にアクセスして取得し、検索結果を出力装置4に出力する(ステップB5)。

【0067】ステップB7において、アトリビュートのインデックスを更新する際の動作について詳細に説明する。前述のように、アトリビュートインデックス記憶部32は、図18に示す社員の名前と年齢とをキーとしたマルチキーインデックスを記憶している。フィルタ条件は、名前がAからKで始まることと年齢が30歳以下であることとする。

【0068】マルチキーインデックスでは、ブリックaから探索が開始される。ブリックaでは、ルートにおいて「A」および「K」とx3(S)とが比較され、左の枝が選択される。続いて、次のノードで「30歳以下」とy1(28)とが比較され、ここでは両者の大小関係が不明なので両方の枝が選択される。したがって、ブリックaでは、ブリックbとブリックcの両方が選択される。

【0069】ブリックbでは、ルートにおいて、「A」および「K」とx2(1)とが比較され、ここでは両者の関係が不明なので両方の枝が選択される。したがって、ブリックbでは、リーフD、Eが選択される。

【0070】ブリックbでは、ルートにおいて「A」および「K」とx3(S)とが比較され、左の枝が選択される。続いて、次のノードで「30歳以下」とy1(28)とが比較され、ここでは両者の大小関係が不明なので両方の枝が選択される。しかし、左の枝には外部ノード(ext)が指定されているので、左の枝についての探索はここでストップする。右の枝については、「A」および「K」とx1(A)とが比較され、リーフBが選択される。そして、選択されたリーフB、D、Eに登録されたエントリの中から、フィルタ条件を満たす(Ao

k i、32)が抽出され、この(A o k i、32)のエントリのDNがフィルタ検索手段23に出力される。

【0071】本実施形態のディレクトリ検索システムでは、各エントリのアトリビュートに基づいて構成されるインデックスが、h B t r e e方式のマルチキーインデックスであるため、複数のアトリビュートがフィルタ条件として指定されたときでも、複数のインデックスを保持する必要がなくなる。そのため、本実施形態のディレクトリ検索システムでは、インデックスを記憶するための記憶容量を小さくすることができる。

【0072】また、本実施形態のディレクトリ検索システムでは、前述のように各エントリのアトリビュートに基づいて構成されるアトリビュートのインデックスが、複数のアトリビュートをキーとするh B t r e e方式のマルチキーインデックスとなっている。そのため、本実施形態のディレクトリ検索システムでは、複数のアトリビュートがフィルタ条件として指定されたときでもインデックスの探索を1回で済ませることができる。そのため、本実施形態のインデックスシステムでは、複数のアトリビュートを探索する場合に複数のインデックスを検索したり何度も同じエントリにアクセスする必要がなくなるため、エントリの検索時間を短縮することができる。

【0073】(第2の実施形態)次に、本発明の第2の実施形態のディレクトリ検索システムについて説明する。図4は、本実施形態のディレクトリ検索システムの構成を示すブロック図である。図4を参照すると、本実施形態のディレクトリ検索システムは、入力装置1と、データ処理装置5と、記憶装置6と、出力装置4とから構成されている。データ処理装置5は、図1に示すデータ処理装置2が備えるフィルタ検索手段23とスコープ判定手段24とが、スコープ/フィルタ統合検索手段25に置き換えられたものである。また、記憶装置6は、図1に示す記憶装置3の構成要素に加え、先祖関係記憶部34をさらに備えている。

【0074】アトリビュートインデックス記憶部32は、図5に示すh B t r e e形式のマルチキーインデックスを備える。このマルチキーインデックスは、ノード501~503と、リーフ204~208とから構成される。ノード501~ノード503は、それぞれがブリックa、b、cである。ブリックa、b、c内のインデックスの各ノードには、各ノードに格納されているキーの値を有するエントリの階層番号および階層別番号が付与されている。

【0075】スコープ/フィルタ統合検索手段25は、マルチキーインデックス管理手段22を介して指定されたフィルタ条件に基づいてアトリビュートインデックス記憶部32に記憶されているマルチキーインデックスの探索を行うとともに、スコープ条件が指定されている場合には先祖関係記憶部34が記憶する先祖関係表を用い

てスコープ判定も同時に行う。また、スコープ/フィルタ統合検索手段25は、スコープ判定を行った結果を記録するためのスコープ表を備える。図6は、本実施形態のディレクトリ検索システムにおけるスコープ/フィルタ統合検索手段25が備えるスコープ表である。スコープ表の各行は、図20のディレクトリ階層の各階層番号を表しており、スコープ表の各列は、スコープ左外(L e f t - O u t、以降L - O)、スコープ右外(L e f t - I n、以降L - I)、スコープ右内(R i g h t - I n、以降R - I)、スコープ右外(R i g h t - O u t、以降R - O)とに区切られている。

【0076】エントリの階層別番号がL - O、R - Oに書き込まれているということは、そのエントリはスコープ条件の範囲外にあることを意味している。また、エントリの階層別番号がL - I、R - Iに書き込まれているということは、そのエントリはスコープ条件の範囲内にあることを意味している。

【0077】本実施形態のディレクトリ検索システムの動作を図7、図8を参照して詳細に説明する。図7は、エントリやアトリビュートの更新要求があった場合のディレクトリ検索システムの動作を示すフローチャートである。なお、図7におけるステップA1、A4~A7の動作は、図2におけるエントリ管理手段21の動作と同一のため、説明は省略する。

【0078】ステップA1において、利用者からの要求がエントリの更新である場合、エントリ管理手段21は、マルチキーインデックス管理手段22を介してエントリインデックス記憶部31に記憶されるエントリの更新を行う(ステップA2)。そして、エントリ管理手段21は、先祖関係記憶部34のエントリの更新にともなう変化したエントリの先祖関係に基づいて先祖関係表を更新する(ステップC1)。その後、エントリ管理手段21は、エントリ/アトリビュート記憶部33に記憶されるエントリの更新を行う(ステップA3)。

【0079】図8は、エントリの検索を行う場合の本実施形態のディレクトリ検索システム動作を示すフローチャートである。ステップB1、B2、B6、B9、B10における動作は、図3に示す動作と同一であるため、これらのステップにおける動作の説明は省略する。

【0080】ステップB3で検索要求にフィルタ条件が含まれており、ステップB6でフィルタ条件となっているアトリビュートにインデックスが付与されている場合には、スコープ/フィルタ統合検索手段25は、検索要求にスコープ条件が含まれているかを調べる(ステップD1)。

【0081】もし、検索要求にスコープ条件が含まれている場合、スコープ/フィルタ統合検索手段25は、マルチキーインデックス管理手段22を介してフィルタ条件に基づいてアトリビュートインデックス記憶部32が記憶するインデックスを探索するとともに、インデック

ス探索途中で先祖関係記憶部34の先祖関係表を参照してインデックスの各ノードの分岐条件となっているアトリビュート値を有するエントリがスコープの範囲内にあるエントリであるかどうかを調べ、スコープ表を作成する(ステップD2)。ステップD2においてフィルタ条件を満たすエントリを検索後、それらのエントリの中からスコープ条件を満たすエントリをスコープ表から求める。スコープ表からスコープ条件を満たしているかどうか判断できないエントリについては、先祖関係記憶部34の先祖関係表を参照してスコープ判定を行う(ステップD3)。

【0082】ステップD1においてスコープ条件が含まれていない場合には、第1の実施形態のディレクトリ検索システムと同様にアトリビュートインデックス記憶部32のインデックスを探索のみを行う(ステップB7)。ステップD3またはステップB7の終了後、ステップD3またはステップB7において検索されたエントリの階層番号および階層別番号をエントリインデックス記憶部31のインデックスを探索して得る(ステップB8)。そして、エントリ／アトリビュート記憶部33から検索結果のエントリの管理情報を取得する(ステップD4)。

【0083】また、ステップB4において、検索要求にスコープ条件が含まれている場合には、スコープ／フィルタ統合手段25は、先祖関係記憶部34の先祖関係表を参照してスコープの判定を行う(ステップD3)。ステップD3、B4、D4終了後、スコープ／フィルタ統合手段25は、エントリ／アトリビュート記憶部33にアクセスして最終的な検索結果のエントリの管理情報を取得し、出力装置4にそれらの管理情報を出力する(ステップB5)。

【0084】アトリビュートインデックス記憶部32の記憶するマルチキーインデックスでは、前述のように、ブリック501～503のインデックスのノードには、各ノードの分岐条件となっているキー値をアトリビュートとして有するエントリの階層番号および階層別番号が付与されている。

【0085】ステップD2において、スコープ／フィルタ統合検索手段25は、フィルタ条件となっているアトリビュートをキーとするインデックスがアトリビュートインデックス記憶部32に存在している場合に、マルチキーインデックス管理手段22を介してアトリビュートインデックス記憶部32に記憶されるインデックスを探索する。そして、各ブリックのインデックスのノードを通過する際に、そのノードに付与されている階層番号および階層別番号を有するエントリが、スコープ条件の範囲内かどうかを先祖関係記憶部34の先祖関係表を参照して調べ、その結果に基づいてスコープ表を更新する。

【0086】例えば、フィルタ条件が年齢30歳以上であることであり、スコープ条件が営業部門であるとし

る。スコープ／フィルタ統合検索手段25は、アトリビュートインデックス記憶部32のマルチキーインデックスを年齢30歳以上で探索する。スコープ／フィルタ統合検索手段25は、その探索途中でノードを通過する際に、そのノードに付与されている階層番号および階層別番号を有するエントリがスコープ条件となっている営業部門に属しているかどうかをスコープ表にチェックしていく。階層番号4および階層別番号1のSat o氏と階層番号4および階層別番号2のEnd o氏とは、営業部門に属しているので、スコープ条件の範囲内にある。したがって、スコープ／フィルタ統合検索手段25は、スコープ表の階層番号4およびL-1、R-1のところに階層別番号1、2を書き込む。さらに、階層番号4および階層別番号5のA o k i氏は、営業部門に属していないので、スコープ条件の範囲外である。したがって、スコープ／フィルタ統合検索手段25は、スコープ表の階層番号4でR-0のところに階層別番号5が書き込む。以上の結果から、階層番号4における各エントリのうち、階層別番号が1、2であるエントリはスコープ条件の範囲内であり、階層別番号が5であるエントリはスコープ条件の範囲外であることがわかる。

【0087】フィルタ条件によるマルチキーインデックスの探索は、最終的にリーフA、B、Cが選択され、そこに登録されているエントリから30歳以上のA o k i氏、(階層別番号5)、N a k a i氏(階層別番号8)、U e d a氏(階層別番号3)が抽出される。この中からスコープ条件の範囲外である階層別番号5のA o k i氏は検索結果から除外される。

【0088】本実施形態のディレクトリ検索システムでは、フィルタ条件を満たすエントリの検索中にスコープの絞り込みを行うことで、スコープ判定の負担を軽減することができる。また、本実施形態のディレクトリ検索システムでは、フィルタ検索とスコープ判定の処理を統合することにより、結果的にスコープ判定に含まれないエントリにアクセスすることがなくなるため、エントリの検索時間を短縮することができる。

【0089】(第3の実施形態)次に、本発明の第3の実施形態のディレクトリ検索システムについて説明する。図9は、本実施形態のディレクトリ検索システムの構成を示すブロック図である。本実施形態のディレクトリ検索システムは、入力装置1と、出力装置4と、記憶装置6と、データ処理装置7とから構成されている。データ処理装置7は図4のデータ処理装置5の構成要素に加えて、部分文字列操作手段26を備えている。部分文字列操作手段26は、マルチキーインデックス管理手段22からの指示により、文字列を分解する。

【0090】本実施形態のディレクトリ検索システムでは、名前の部分一致検索を行うため、各社員の名前から抽出された3文字の部分文字列をキーとする。図10は、図20のディレクトリ階層における社員の名前から

抽出された3文字の部分文字列と年齢とから成るマルチキーの分布状態を示すマップである。このマップは、縦軸が年齢で、横軸が名前の先頭のアルファベットとなっている。

【0091】図10に示すように、アルファベットSをx3とし、アルファベットIをx2とし、アルファベットEをx1とし、年齢28をy1とする。マップ上のエリアは、年齢28以上かつ名前の頭文字がA~Dの領域701と、年齢28以上かつ名前の頭文字がE~Rの領域702と、年齢28未満で名前の頭文字がA~Hの領域703と、年齢28未満で名前の頭文字がI~Rの領域704と、名前の頭文字がS~Zの領域705とに分割されている。

【0092】図11は、本実施形態のディレクトリ検索システムにおけるアトリビュートインデックス記憶部32のマルチキーインデックスを示す図である。このマルチキーインデックスは、ノード601~603と、リーフ604~608とから構成されている。ノード601~603は、それぞれがブリックa、ブリックb、ブリックcである。図11のhB treeでは、図10におけるマップの各領域を元にブリックa、b、cを構成する。ブリックaは、マップ全体を指すものであるとし、ブリックcは、領域701、702、705を指すものであるとし、ブリックbは、領域703、704を指すものであるとする。

【0093】このマルチキーインデックスは、まずフィルタ条件として指定されたアトリビュートの値が、マップ上のどの領域の属するかを探索する。ブリックaでは、x3(S)とy1(28)とを分岐条件として、フィルタ条件として指定されたアトリビュートの値が、領域701、702、705に含まれるか領域703、704に含まれるかを切り分ける。そして、ブリックbでは、フィルタ条件として指定されたアトリビュートの値の含まれる領域が、領域703なのか領域704なのかを切り分ける。ブリックcでは、フィルタ条件として指定されたアトリビュートの値の含まれる領域が、領域701なのか領域702領域705なのかを切り分ける。

【0094】リーフ604~608にはそれぞれ領域703、704、701、702、705に含まれる各要素のDNが登録されている。同一のリーフ内に格納され、同じエントリのDNを指す要素は、マルチキーインデックス管理手段22によってキー併合されて登録されている。例えば、各マルチキー(I I J、20)、(I J I、20)、(J I M、20)、(I M A、20)は、同じリーフ608に格納され、同じエントリであるI i j i m a氏を指すので(I I J+I J I+J I M+I M A、20)というふうにマルチキーインデックス管理手段22によってキー併合されて登録されている。

【0095】次に、本実施形態のディレクトリ検索システムの動作を図12、図13を参照して詳細に説明す

る。図12は、エントリまたはアトリビュートの更新を行う場合の本実施形態のディレクトリ検索システムの動作を示すフローチャートである。図12のステップA1~A5、A7、C1の動作は、第2の実施形態のディレクトリ検索システムにおけるエントリ管理手段21の動作と同一であるため、これらのステップの動作の説明は省略する。

【0096】ステップA5において、利用者からの要求がインデックスが付与されているアトリビュートの更新要求であった場合、マルチキーインデックス管理手段21は、アトリビュートに文字列が含まれているかを調べる(ステップE1)。

【0097】もし含まれていれば、マルチキーインデックス管理手段22は、部分文字列操作手段26を介して文字列から3文字の部分文字列を抽出する(ステップE2)。例えば、更新要求のあったアトリビュートの値が、(T s u j i、30)であった場合、文字列「t s u j i」から「t s u」と「s u j」と「u j i」を抽出する。すると、マルチキーインデックス管理手段22は、(t s u、30)、(s u j、30)、(u j i、30)の3つのマルチキーを作成し、これらのマルチキーによってアトリビュートインデックス記憶部のマルチキーインデックスを探索する。そして、各マルチキーのうち、探索結果が同一のリーフとなったマルチキーがある場合、マルチキーインデックス管理手段22は、それらのマルチキーのキー併合を行いアトリビュートインデックス記憶部32のインデックスを更新する(ステップE3)。例えば、上述のマルチキー(t s u、30)、(s u j、30)、(u j i、30)は、すべて、リーフ608に格納されるようになるので、(t s u+s u j+u j i、30)のように、キー併合を行って(t s u j i、30)のエントリのDNをリーフ608に格納する。

【0098】その後、マルチキーインデックス管理手段22は、エントリ/アトリビュート記憶部33に新しく追加されたエントリのアトリビュートを更新する(ステップA7)。ステップE1において、指定されたアトリビュートに部分文字列が含まれていなかった場合、マルチキーインデックス管理手段は、そのアトリビュートに基づいて、アトリビュートインデックス記憶部32に格納されたインデックスを更新し(ステップA6)、ステップA7に進む。

【0099】図13は、エントリの検索要求があった場合の本実施形態のディレクトリ検索システムの動作を示すフローチャートである。図13において、ステップB1~B10、D1~D4の動作は、図8におけるそれらのステップの動作と同一であるため、それらのステップの動作の説明は省略する。

【0100】ステップB3でフィルタ条件が指定されており、ステップB6でインデックスが付与されていた場

合、マルチキーインデックス管理手段22は、部分文字列のインデックスが含まれているかを調べる（ステップF1）。もし含まれていれば、部分文字列操作文字手段26は、その部分文字列から所定の文字数の部分文字列を抽出する（ステップF2）。例えば、「NAKA」という文字列を含む名前を持つエントリを検索する場合、所定の文字数が3文字であるとする、部分文字列操作手段26は「NAKA」から「NAK」および「AKA」を抽出する。その後、本実施形態のインデックスシステムの動作は、スコープ／フィルタ統合検索手段25に移行する。ステップF1において部分文字列のインデックスが含まれていない場合には、ステップF2の処理はスキップされる。

【0101】ブリックa、b、c内のインデックスの内部ノードには、第2の実施形態のディレクトリ検索システムと同様にスコープの絞り込みで使用されるエントリの階層番号および階層別番号が付与されている。リーフ604～608には、キー併合により同じエントリを指しているマルチキーがキー併合されて管理されている。「NAK」による探索では、リーフ605と607とにたどりついて（NAK+KAI、37）のエントリが選択される。「AKA」による探索ではリーフ604と606にたどりついて（AKA、37）のエントリが選択される。これらのエントリは、同じNakai氏を示すので、Nakai氏が検索結果となって、出力装置4に出力される。

【0102】また、本実施形態のディレクトリ検索システムでは、「Nakai」のような全文字列検索を行う場合でも、同様な動作で上述のマルチキーインデックスを探索してNakai氏の情報を取得することができる。

【0103】以上述べたように、本実施形態のディレクトリ検索システムでは、文字列から所定の文字数の部分文字列を抽出する部分文字列操作手段26と、すべての部分文字列の検索結果に含まれるエントリを検索結果であると判断するマルチキーインデックス管理手段22とを備えることによって、同一のインデックスで全文字列および部分文字列の検索を行うことができるため、インデックスの記憶に要する記憶容量を少なくすることができる。

【0104】また、本実施形態のディレクトリ検索システムでは、部分文字列をキーとするインデックスについて、同一のリーフに登録される同一のエントリについてのマルチキーがあった場合には、それらのマルチキーをキー併合するため、リーフに登録されるマルチキーの数を減らすことができるため、インデックスを記憶するための記憶容量を小さくすることができる。

【0105】なお、上記第1、第2、第3の実施形態のディレクトリ検索システムでは、アトリビュートインデックス記憶部が格納するインデックスを**h B t r e e**

の形態としたマルチキーインデックスであるとして構造および動作を説明したが、本発明はこれに限定されるものではなく、マルチキーインデックスを**k e e d t r e e**または他の形態のマルチキーインデックスとしても良い。

【0106】また、図14に示すように、第1、第2、第3の実施形態のディレクトリ検索システムは、データ更新またはデータ検索を実行するためのプログラムを記録した記録媒体7を備えていてもよい。

【0107】第1の実施形態のディレクトリ検索システムでは、記録媒体7は、フィルタ条件とを満たすエントリをマルチキーインデックスから検索する処理と、エントリやアトリビュートの登録や削除や更新が行われる場合に、ディレクトリ階層およびマルチキーインデックスに対しても登録や削除や更新を行う処理とを行うプログラムを記録している。

【0108】また、第2の実施形態のディレクトリ検索システムでは、記憶媒体7は、ディレクトリ階層におけるスコープ条件が指定されていた場合にフィルタ条件を満たすエントリをマルチキーインデックスを検索する途中に通過する各ノードに付与されている階層番号および階層別番号を有するエントリがスコープ条件に入っているか否かを各エントリの先祖関係からチェックする処理と、そのチェックの結果においてスコープ条件に含まれていなかったエントリをフィルタ条件を満たすエントリから除外した後で、残りのエントリがスコープ範囲に含まれているか否かの判定を行う処理と、利用者からエントリの登録や削除や更新が行われた場合には、ディレクトリ階層の登録や削除や更新を行うとともにマルチキーインデックスの2分探索木の各ノードに付与されている階層番号および階層別番号の登録や削除や更新を行う処理とを行うプログラムを記録している。

【0109】また、第3の実施形態のディレクトリ検索システムでは、記録媒体7は、検索条件として指定された属性値が文字列であった場合、その文字列の中から所定の文字数の部分文字列をすべて抽出する処理と、フィルタ条件として指定された属性値が文字列であった場合にその文字列から抽出された少なくとも1つの部分文字列によって所定の文字数の部分文字列をキーの1つとするマルチキーインデックスの探索を行い、各部分文字列毎のマルチキーインデックスの探索結果となったエントリのうちすべての部分文字列についての探索結果に含まれるエントリをフィルタ条件を満たすエントリとする処理と、利用者から属性となっている文字列の更新要求があった場合にその文字列から抽出された所定の文字数の部分文字列のうちの探索結果が同一になる部分文字列についてはキー併合を行ってその部分文字列が指すエントリの識別子を格納する場所を同一とする処理とを行うためのプログラムを記録している。記録媒体7としては磁気ディスク、半導体メモリまたはその他の記録媒体が用

いられる。

【0110】

【発明の効果】以上述べたように、本発明のディレクトリ検索システムは、以下に示す3つの効果を有している。

(1) アトリビュートインデックス記憶部が記憶するインデックスが、複数のキーによって一度に探索可能なマルチキーインデックスとなっているため、アトリビュートからエントリを検索するためのインデックスを記憶するのに必要な記憶容量を少なくことができる。また、複数のインデックスを検索する必要がないので、検索時間を短縮することができる。

(2) マルチキーインデックスの探索時にスコープの絞り込みを行い、スコープの判定処理の負荷を軽減しているため、スコープ判定処理の負荷を軽減してエントリの検索時間を短縮することができる。また、フィルタ検索とスコープ判定の処理を統合することにより、結果的にスコープ判定に含まれないエントリにアクセスすることがなくなるため、エントリの検索時間を短縮することができる。

(3) 文字列から所定の文字数の部分文字列を抽出する部分文字列操作手段と、すべての部分文字列の検索結果に含まれるエントリを検索結果であると判断するマルチキーインデックス管理手段とを備えることによって、同一のインデックスで全文字列および部分文字列の検索を行うことができるため、インデックスを記憶するのに必要な記憶容量を少なくすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態のディレクトリ検索システムの構成を示すブロック図である。

【図2】エントリやアトリビュートの更新要求が入力された場合の本発明の第1の実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【図3】エントリを検索する場合の本発明の第1の実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【図4】本発明の第2の実施形態のディレクトリ検索システムの構成を示すブロック図である。

【図5】本発明の第2の実施形態のディレクトリ検索システムにおけるアトリビュートインデックス記憶部が記憶するマルチキーインデックスを示す図である。

【図6】本発明の第2の実施形態のディレクトリ検索システムにおけるスコープ／フィルタ検索手段が記憶するスコープ表を示す図である。

【図7】エントリまたはアトリビュートの更新を行う場合の本発明の第2の実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【図8】エントリの検索を行う場合の本発明の第2の実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【図9】本発明の第3の実施形態のディレクトリ検索システムの構成を示すブロック図である。

【図10】本発明の第3の実施形態のディレクトリ検索システムにおけるアトリビュートインデックス記憶部が記憶するマルチキーインデックスを示す図である。

【図11】ある会社の社員の名前の部分文字列を抽出した場合の各部分文字列と年齢とから構成されるマルチキーの分布を示すマップを示す図である。

【図12】エントリまたはアトリビュートを更新する場合の本発明の第3の実施形態のディレクトリ検索システムの動作を示すフローチャートである。

【図13】本発明の第3の実施形態のディレクトリ検索システムのエントリまたはアトリビュートの検索時における動作を示すフローチャートである。

【図14】本発明の第1、第2、第3の実施形態のディレクトリ検索システムが記録媒体を備えている場合の構成を示すブロック図である。

【図15】特開平3-70049号公報に開示されたディレクトリ検索システムの構成を示すブロック図である。

【図16】k-d treeの基本的な構成を示す図である。

【図17】ある会社の社員の名前と年齢とから構成されるマルチキーの分布状態を示すマップである。

【図18】ある会社の社員の構成をhB treeの形態で表した図である。

【図19】特願平11-43259号出願に記載された従来のディレクトリ検索システムの構成を示すブロック図である。

【図20】ある会社の組織をディレクトリ階層で表す図である。

【図21】従来のディレクトリ検索システムの先祖関係記憶部が記憶する先祖関係表を示す図である。

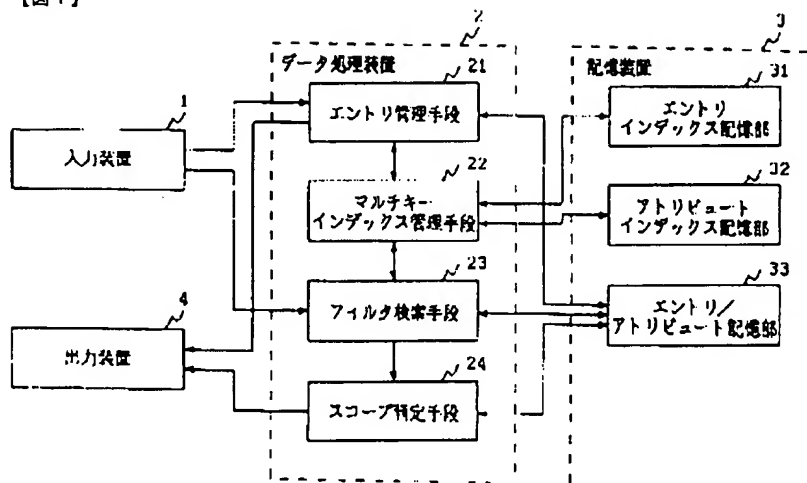
【符号の説明】

- 1 入力装置
- 2、5、7、8 データ処理装置
- 3、6、9 記憶装置
- 4 出力装置
- 21 エントリ管理手段
- 22 マルチキーインデックス管理手段
- 23、27 フィルタ検索手段
- 24、28 スコープ判定手段
- 25 スコープ／フィルタ統合検索手段
- 26 部分文字列操作手段
- 31 エントリインデックス記憶部
- 32 アトリビュートインデックス記憶部
- 33 エントリ／アトリビュート記憶部
- 34 先祖関係記憶部
- 35 複合インデックス記憶部
- 101～108 ノード

204~208、604~608 リーフ
 201~203、501~503、601~603
 ノード
 301~305、701~705 領域
 401 会社
 402 営業部門
 403 開発部門
 404 国内営業課
 405 海外営業課
 406 企画課
 407 製造課
 408~415 社員
 1001 処理開始指示装置
 1002 ディレクトリ情報処理手段

1003 併合インデックス作成指示手段
 1004 併合インデックス読込指示手段
 1005 併合インデックス作成手段
 1006 併合インデックス読込手段
 1007 作業ファイル
 1008 併合インデックス
 1009 ディレクトリファイル
 1010 利用者インデックス
 1011 グループ共有インデックス
 1012 システム共有インデックス
 1013 サブファイル群
 A1~A7、B1~B11、C1 ステップ
 D1~D4、E1~E3、F1、F2 ステップ

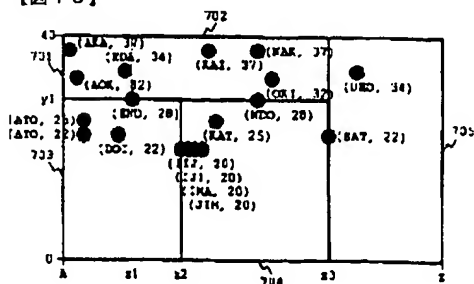
【図1】



【図6】

	L-0	L-1	R-1	2-0
1	—	—	—	—
2	—	—	—	—
3	—	—	—	—
4	—	1	2	5

【図10】



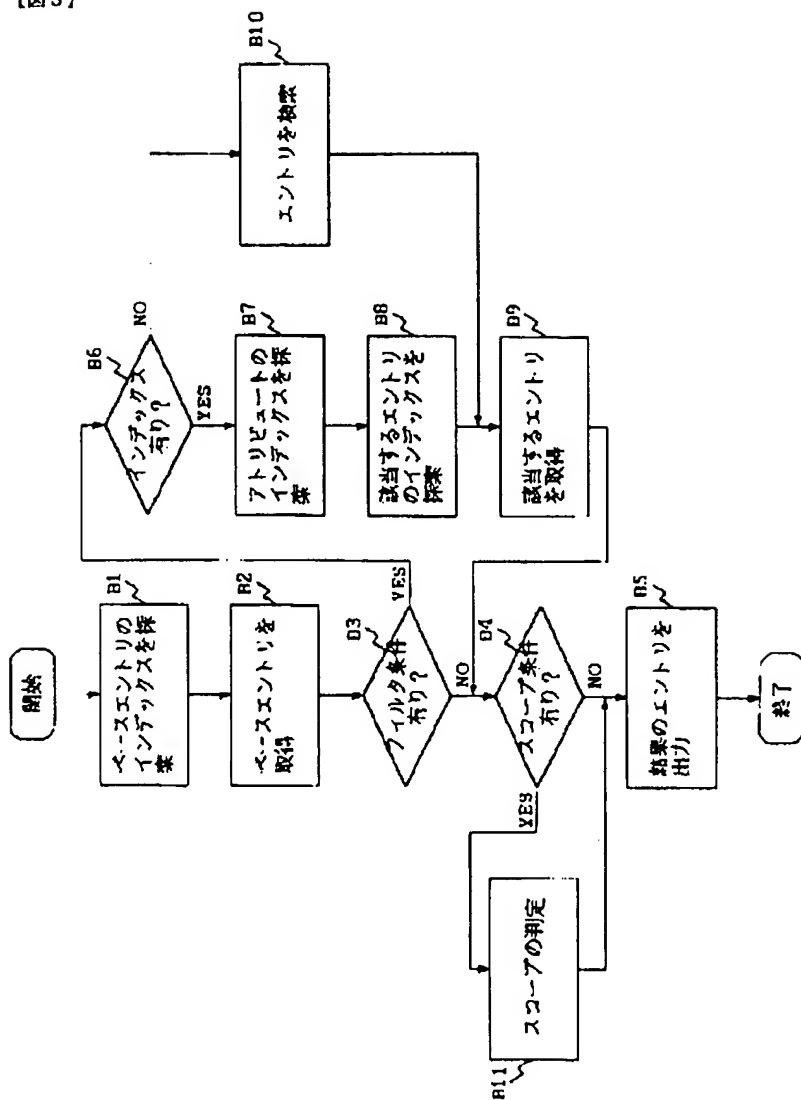
```
graph TD
    Start([開始]) --> A1{エントリ更新?}
    A1 -- YES --> A2[エントリのインデックスを更新]
    A1 -- NO --> A5{インデックス有り?}
    A2 --> A3[エントリを更新]
    A3 --> A4[処理終了を通知]
    A4 --> End([終了])
    A5 -- YES --> A6[アトリビュートのインデックスを更新]
    A5 -- NO --> A3
    A6 --> A7[アトリビュートを更新]
    A7 --> A3
```

The flowchart illustrates the processing procedure for the first embodiment. It begins with a start node (開始), leading to a decision point A1: "エントリ更新?" (Entry update?). If the answer is YES, the process proceeds to A2: "エントリのインデックスを更新" (Update entry index), then to A3: "エントリを更新" (Update entry), and finally to A4: "処理終了を通知" (Notify processing completion), which leads to the end node (終了). If the answer to A1 is NO, the process proceeds to A5: "インデックス有り?" (Index present?). If the answer to A5 is YES, the process proceeds to A6: "アトリビュートのインデックスを更新" (Update attribute index), then to A7: "アトリビュートを更新" (Update attribute), and finally to A3. If the answer to A5 is NO, the process proceeds directly to A3. A dashed line indicates a loop from the point after A3 back to A1.

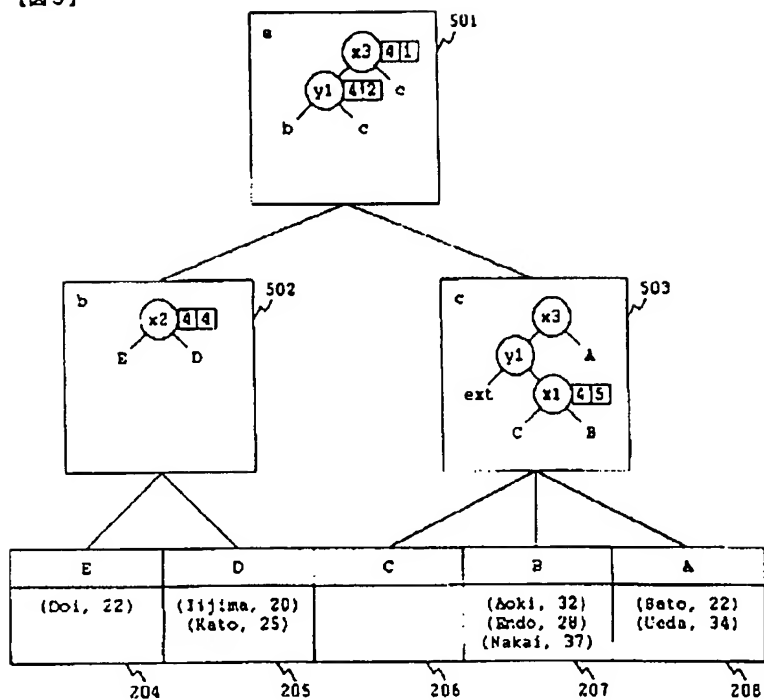
Figure 1 is a hierarchical diagram showing the decomposition of a 32-bit instruction into four 8-bit segments (a, b, c, d) and their corresponding bit fields. The top level is a 32-bit instruction, which is divided into four 8-bit segments. Segment 'a' (bits 31-24) contains fields ATO (bits 31-28), COO (bits 27-24), and AYO (bits 23-20). Segment 'b' (bits 23-16) contains fields ATO (bits 23-20), COO (bits 19-16), and AYO (bits 15-12). Segment 'c' (bits 15-8) contains fields ENO (bits 15-12), COO (bits 11-8), and AYO (bits 7-4). Segment 'd' (bits 7-0) contains fields ENO (bits 7-4), COO (bits 3-0), and AYO (bits 3-0). The diagram illustrates the mapping of instruction bits to these fields.

図4は、データ処理装置と記憶装置の構成を示すブロック図である。データ処理装置（5）は、エントリ管理手段（21）、マルチキーインデックス管理手段（22）、およびスコア/フィルタ統合検索手段（25）を含む。記憶装置（6）は、エントリインデックス記憶部（31）、アトリビュートインデックス記憶部（32）、エントリ/アトリビュート記憶部（33）、および先祖関係記憶部（34）を含む。入力装置（1）はエントリ管理手段（21）とスコア/フィルタ統合検索手段（25）に接続されている。出力装置（4）はエントリ管理手段（21）とスコア/フィルタ統合検索手段（25）からデータを受け取る。エントリ管理手段（21）はマルチキーインデックス管理手段（22）とエントリインデックス記憶部（31）と接続されている。マルチキーインデックス管理手段（22）はスコア/フィルタ統合検索手段（25）とアトリビュートインデックス記憶部（32）と接続されている。スコア/フィルタ統合検索手段（25）はエントリ/アトリビュート記憶部（33）と先祖関係記憶部（34）と接続されている。また、エントリインデックス記憶部（31）はエントリ管理手段（21）と接続されている。アトリビュートインデックス記憶部（32）はマルチキーインデックス管理手段（22）と接続されている。エントリ/アトリビュート記憶部（33）はスコア/フィルタ統合検索手段（25）と接続されている。先祖関係記憶部（34）はスコア/フィルタ統合検索手段（25）と接続されている。

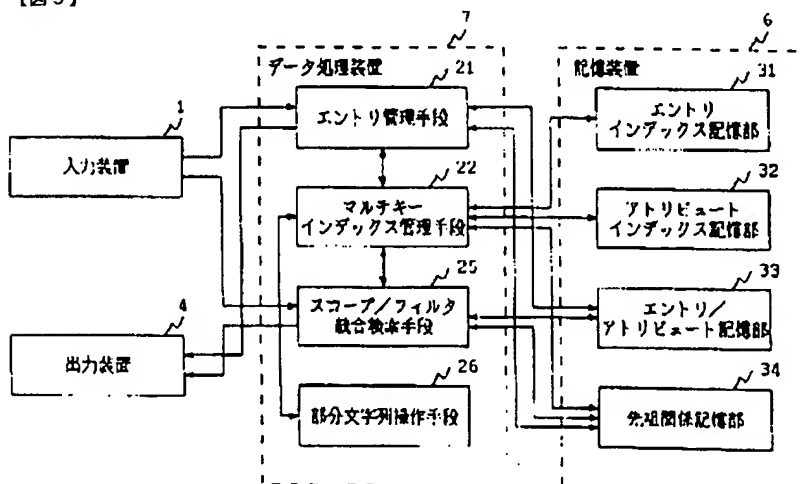
【図3】



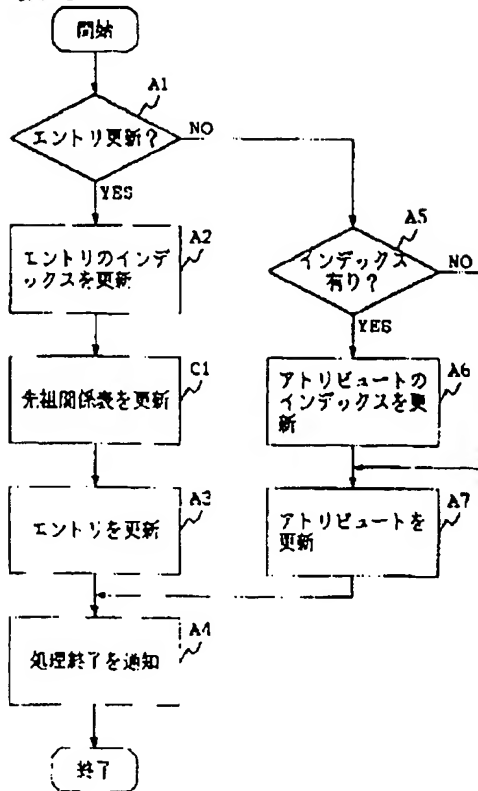
【図5】



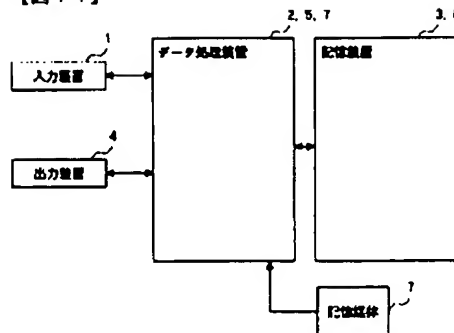
【図9】



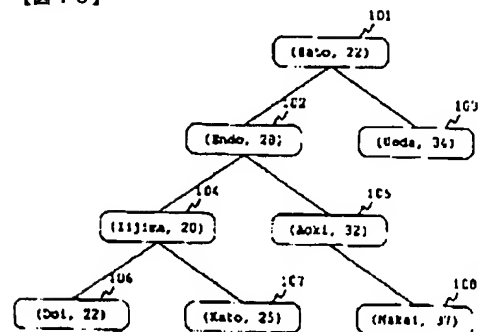
【図 7】



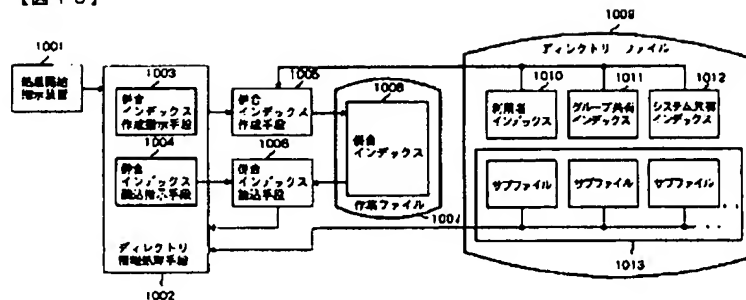
【図 14】



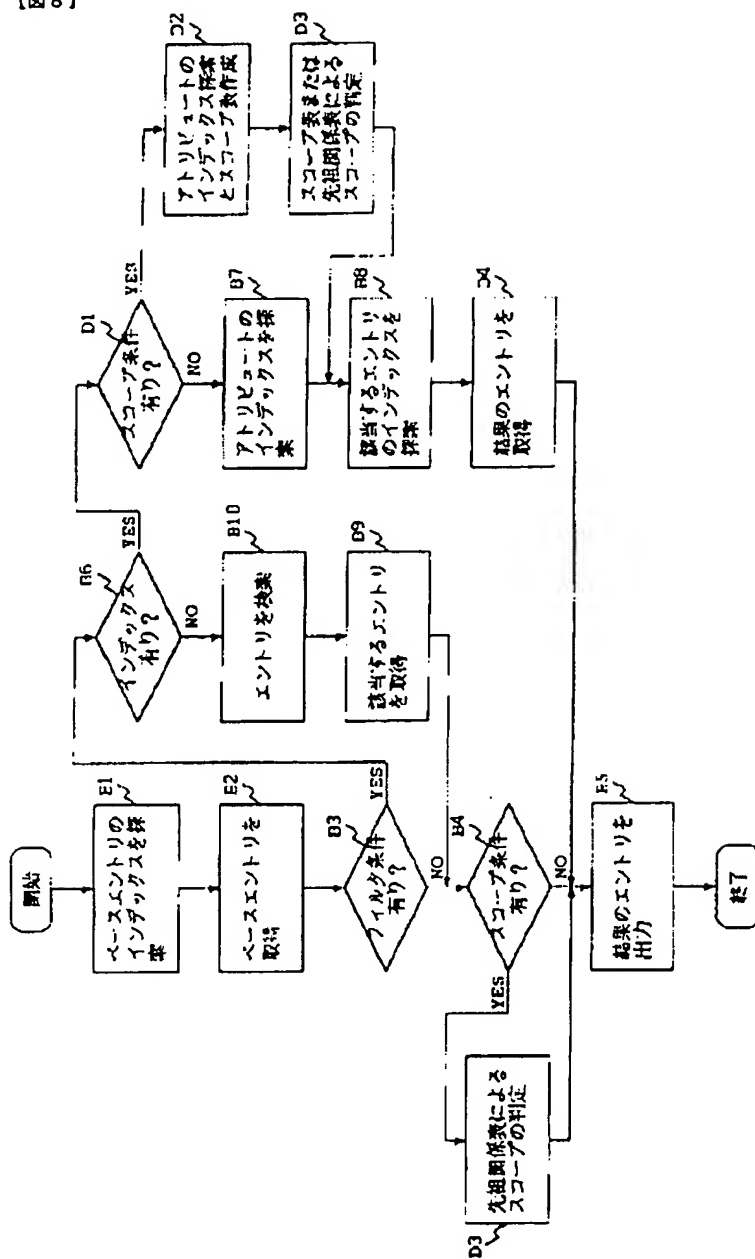
【図 16】



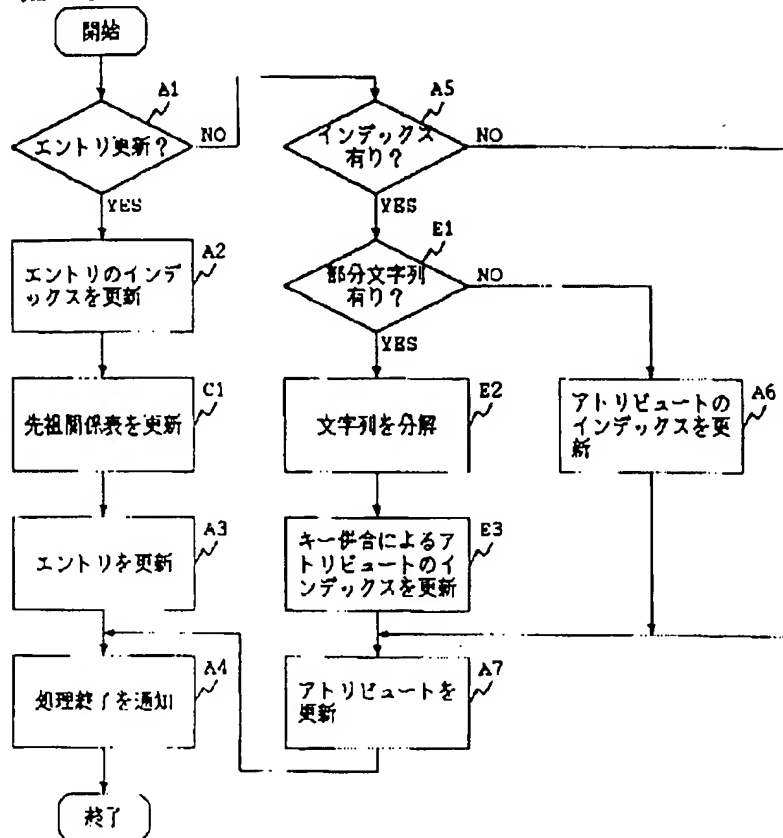
【図 15】



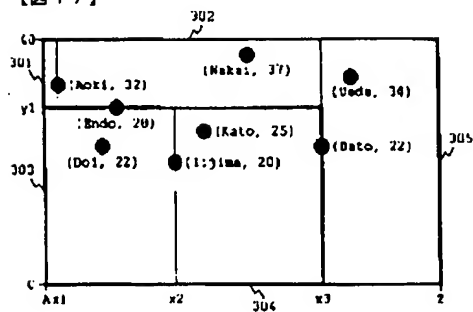
【図8】



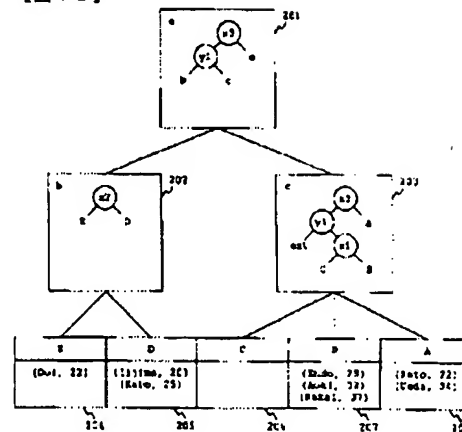
【図 12】



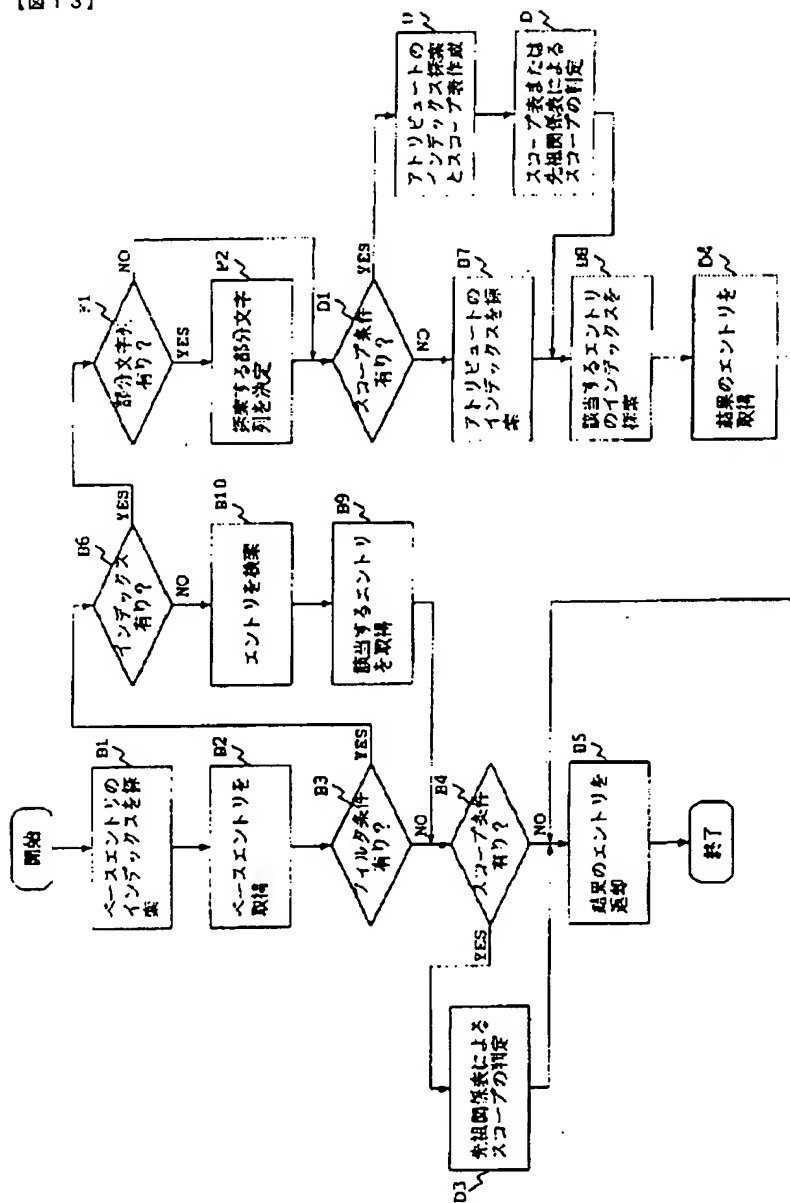
【図 17】



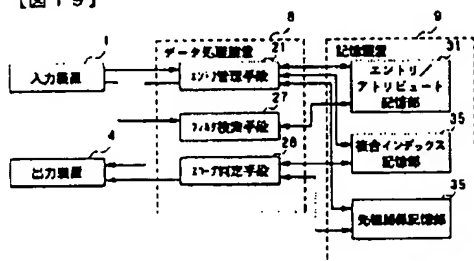
【図 18】



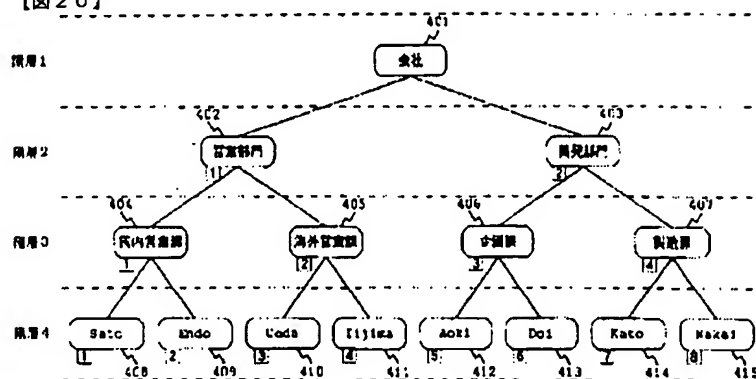
【図13】



【図19】



【図20】



【図21】

		1	2		3			
		会社	営業部門	開発部門	国内営業課	海外営業課	企画課	製造課
2	営業部門	✓						
	開発部門	✓						
3	国内営業課	✓	✓					
	海外営業課	✓	✓					
	企画課	✓		✓				
	製造課	✓		✓				
4	Sato	✓	✓		✓			
	Endo	✓	✓		✓			
	Ueda	✓	✓			✓		
	Iijima	✓	✓			✓		
	Aoki	✓		✓			✓	
	Doi	✓		✓			✓	
	Kato	✓		✓				✓
	Nakai	✓		✓				✓